Real-Time Event-Triggered Object Tracking in the Presence of Model Drift and Occlusion

Guan, Mingyang; Wen, Changyun; Shan, Mao; Ng, Cheng-Leong; Zou, Ying

2018

Guan, M., Wen, C., Shan, M., Ng, C.-L., & Zou, Y. (2018). Real-Time Event-Triggered Object Tracking in the Presence of Model Drift and Occlusion. IEEE Transactions on Industrial Electronics, in press.

<https://hdl.handle.net/10356/88995>

<https://doi.org/10.1109/TIE.2018.2835390>

© 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at: [<http://dx.doi.org/10.1109/TIE.2018.2835390>].

Downloaded on 26 Apr 2023 19:38:29 SGT

Real-Time Event-Triggered Object Tracking in the Presence of Model Drift and Occlusion

Mingyang Guan, Changyun Wen, *Fellow, IEEE*, Mao Shan, *Member, IEEE*, Cheng-Leong Ng and Ying Zou

Abstract—In the paper, we propose a novel event-triggered tracking (ETT) framework for fast and robust visual tracking in the presence of model drift and occlusion. The resulting tracker not only operates at real-time, but also is resilient to tracking failures caused by factors such as fast motion and heavy occlusion. Specifically, the tracker consists of an event-triggered decision model as the core module that coordinates other functional modules, including a short-term tracker, occlusion and drift identification, target re-detection, short-term tracker updating and on-line discriminative learning for detector. Each functional module is associated with a defined event that is triggered when a set of proposed conditions are met. The occlusion and drift identification module is intended to perform on-line evaluation of the short-term tracking. When a model drift event occurs, the target re-detection module is activated by the event-triggered decision model to relocate the target and reinitialize the short-term tracker. The short-term tracker updating is carried out at each frame with a variable learning rate depending on the degree of occlusion. A sampling-pool is constructed to store discriminative samples that are used to update the detector model. Extensive experiments on large benchmark datasets demonstrate that the proposed tracking algorithm can effectively detect model drift and restore tracking, and more importantly, it outperforms the state-of-the-art approaches in terms of accuracy, efficiency, and robustness.

Index Terms—visual tracking, drift identification, event-triggering, re-detection, machine learning.

I. INTRODUCTION

Long-term visual object tracking is an important problem in computer vision, with a wide range of applications such as in human-computer interaction, automated surveillance, robot navigation and many others. Great progress has been made in recent years [1]–[8]. Yet numerous practical factors, such as heavy occlusion, appearance variation, illumination variation, abrupt motion, deformation and out-of-view, can easily lead to model drift and thus they need to be taken into account.

The long-term visual tracking is a complicated yet systematic task, which cannot be solved independently through either

tracking or detection. Several interesting trackers are proposed recently to address the long-term tracking problem, for instance, in [9] and [10]. In [9] an on-line random forest detector is proposed and in [10] short-term and long-term target appearance memories are stored to estimate the object location in every frame, and corrects the tracker if necessary. However, detection on each frame is a complex and time-consuming task as the detector needs to search a large area to find the target candidate with the highest response, while tracking usually predicts approximate location of the target with *priori* information and searches the target within a small neighboring region. In many applications, rapid visual tracking is essential to ensure required system performances. For instance, in a scenario where an autonomous robot is to follow a given target, the planned path based on the tracked target is used as a reference trajectory for controlling robot and should be available promptly for implementing control actions, which requires the target to be tracked real-time in various conditions. On the other hand, it should also be noted that detection occurring at each frame is redundant in many cases. Actually, there is little change between two consecutive frames when the time interval is small and in this case just using tracking model without detection can accurately track the target [1]. Thus it is unnecessary to carry out detection at each frame, since short-term tracking algorithm could accurately track the target in most scenarios.

Recently in [11], a tracker named LCT is proposed to activate object re-detection in case of model drift. However, the moment of activation directly relies on the comparison of the confidence of current prediction with a predefined threshold, which could lead to unnecessary triggering of its re-detection module in many cases and thus increase computational cost. Thus, how to identify the model drift accurately is a critical problem, as object detection and model updating at right time will significantly improve the tracking accuracy and also accelerate the tracking speed. In [12] an approach named ROT is proposed to identify the occurrence of occlusion for target patch, and if so, an optimal classifier is chosen from the classifier-pool in terms of entropy minimization. However, occlusion does not mean tracking failure in many scenarios and most short-term trackers are able to tolerate certain level of occlusion except for heavy or long-term partial occlusion. Furthermore, directly replacing current tracking model with the optimal classifiers trained beforehand increases the probability of tracking failure, since it loses the latest frame-to-frame translation information, which is important to short-term tracking. In this case, it is preferable to weaken the weight of noisy samples so as not to contaminate the tracking model, by learning samples discriminatively via excluding heavy occlusion/background samples and decreasing the learning rate

Manuscript received October 27, 2017; revised February 18, 2018 and March 27, 2018; accepted April 27, 2018. The work was partially supported by ST Engineering-NTU Corporate Lab through NRF Corporate Laboratory@University scheme.

M. Guan, C. Wen and Y. Zou are with the School of Electrical and Engineering, College of Engineering, Nanyang Technological University, Singapore (e-mail: myguan@ntu.edu.sg; ecywen@ntu.edu.sg; zouy0011@e.ntu.edu.sg).

M. Shan is with the Australian Centre for Field Robotics, The University of Sydney, Australia (e-mail: seamus_shan@hotmail.com).

C. Ng is with Singapore Technologies Dynamics Pte Ltd, Singapore (e-mail: ngclg@stengg.com)

for partial occlusion.

Motivated by the above discussions and inspired by the idea of event-triggered control proposed in control engineering [13], [14], an event-triggered tracking (ETT) framework with an effective occlusion and model drift identification approach by measuring the temporal and spatial tracking loss is proposed in this paper, in order to simultaneously ensure fast and robust tracking with the required accuracy. A flowchart of the proposed tracker is given in Figure 1. Note that the proposed tracking algorithm is not simply to employ short-term tracker interleaved with detection model. Also as seen from the detail comparisons presented in Table I, the proposed approach has distinguishing features compared with existing drift-alleviating algorithms, which are reviewed and discussed in Section II.

The proposed tracker is mainly composed of the following five modules: a short-term tracker, occlusion and drift identification, target re-detection, short-term tracking model update and on-line discriminative learning for detector. An event-triggered decision model is built as the core component that coordinates these five modules. The short-term tracker is used to efficiently locate the target based on transformation information among the continuous frames. The occlusion and drift identification module evaluates current tracking state. If model drift is detected, the corresponding event will trigger the re-detection module to detect and reinitialize short-term tracking model. The tracker model updating is carried out at each frame, while the learning rate is dependent on the degree of occlusion. A sampling-pool is constructed to store discriminative samples and use them to update detector model.

The main features of the proposed approach and contributions of this paper are summarized as follows.

- A systematic ETT framework is built by decomposing the visual tracker into multiple event-triggered modules which work independently and are driven by particular events. It also allows future upgrade so as to introduce more sophisticated models.
- An occlusion and drift detection method is proposed by considering the temporal and spatial tracking loss. It enables the event-triggered decision model to accurately evaluate the short-term tracking performance, and to trigger the relevant module as needed.
- A bias alleviation sampling approach for SVM model is proposed. A sampling-pool with support samples, high confidence samples, and re-sampling samples, is constructed to alleviate the influence of noisy or mislabeled samples.
- Weighted learning for short-term tracking model is applied, which will learn partial occlusions weakly and reject the heavy occlusion samples completely.

With such a novel event-triggered tracking (ETT) framework, the tracker is able to obtain a more robust short-term tracking performance and also properly address model drift problem for long-term visual tracking. Our proposed scheme is tested on frequently used benchmark datasets TB-100 [15]. From the experimental results presented in Section V, the proposed tracking scheme yields significant improvements over the state-of-the-art trackers under various evaluations conditions. More importantly, the proposed tracker runs at real-time and is much faster than most of the online trackers such as those in [3], [6], [10], [11], [16], [17].

II. PROBLEM FORMULATION AND RELATED WORK

Visual tracking has been studied extensively with numerous applications [18]. In this section, we briefly review some methods closely related to this work and formulate our problems.

A. Trackers with Drift Alleviation/Re-detection

To alleviate the drift problem, some approaches have been proposed by detecting occlusion in order to avert poor model updating or detection tracking failure to re-detect the target and thus reinitialize tracking model. Ma *et al.* [11] use a discriminative correlation filter to estimate the confidence of current tracking state to detect tracking failures and train an online learned random forest classifier to re-detect the target. Further improvement is achieved by maintaining long-term and short-term memory of target appearance to restore the tracker when tracking failure occurs [19]. Dong *et al.* [12] propose a classifier-pool to identify whether the current tracking state is occlusion or not, and an appropriate tracking strategy is proposed for each different states. Yang *et al.* [20] propose to a occlusion sensitive tracking approach by discovering orthogonalized templates, which decompose the residual term into two components in observation model to deal with occlusion cases. Zhang *et al.* [21] propose a discriminative feature selection method which couples the classifier score with the sample importance to enhance the robustness for visual tracking.

Other approaches alleviate the drift problem through detecting the target at each frame to compensate for the drift produced by tracking model. Kalal *et al.* [9] decompose the tracking task into tracking, learning, and detection where tracking and detection complement each other to alleviate the drift problem. Yu *et al.* [22] divide the multi-object tracking task into four processes (*i.e.* active, inactive, tracked and lost processes, respectively). Then a Markov Decision Process is introduced to estimate the state transition. Hong *et al.* [10] introduces a biologic-based model to maintain a correlation filter-based short-tracker and a long-term memory of SIFT key-points for detecting the target.

Generally, the tracking-learning-detection frameworks in [10] and [22] consume substantial computation resources since the re-detection is carried on each frame, while [12] and [11] cannot detect model drift correctly or relocate the target effectively when heavy occlusion or out-of-view happens. In this paper, an occlusion and drift identification model is proposed to trace the tracking failure and thus re-detect the target, in order to avoid these problems.

B. Correlation Filter Based Tracking

Recently, correlation filters have been widely used in visual tracking [1]–[3], [5], [10], [11]. The correlation filter-based trackers can train a discriminative correlation filter efficiently based on the property of the circulant matrix in the frequency domain to achieve satisfactory tracking performance in computational efficiency and accuracy. Bolme *et al.* [2] initially apply correlation filter to visual tracking by minimizing the total squared error on a set of gray-scale patches. Henriques *et al.* [23] improves the performance by exploiting the circulant structure of adjacent image patches to train the correlation filter. Further improvement is achieved by proposing Kernelized Correlation Filters using kernel-based training and HOG features [1]. Danelijan *et al.* [5] propose adaptive multi-scale correlation filters to

Fig. 1. A framework of the proposed tracking algorithm. The correlation tracker is used to track the target efficiently. The tracker records the past predictions to discriminate the occlusion and model drift. The event-triggering model is used to produce triggering signals, labeled in different colors, to activate the corresponding subtasks, respectively. The detection model is used to re-detect the target when model drift happens.

cover the scale change of the target. However, these correlation filter-based trackers are sensitive to long-term occlusion or out-of-view problems as they update tracking model without considering whether the current prediction is accurate or not [1], [5]. To address this limitation, our proposed tracker updates the short-term tracking model discriminatively by filtering out the heavy occlusion/background samples and decreasing the learning rate for partial occlusion.

C. Deep Learning Based Tracking

Deep learning models have been widely explored for visual tracking problem and achieve favorable performance [24]–[26]. Zhang *et al.* [26] formulate the tracking problem as a generative model, where the tracking results are directly employed as the filters in the convolutional layer of CNN. Tao *et al.* [24] utilize a Siamese network to estimate the similarities between the previous target and the candidate patches. Wang *et al.* [25] propose a sequential tracking method of convolutional neural networks for visual tracking, which utilizes an ensemble strategy to avoid over-fitting the network. Generally, the deep learning based tracker can achieve top performance in accuracy on many dataset such as VOT [27], TB-100 [15], but the tracking speed is still slow due to the high dimension of the entire deep feature space. To accelerate the tracking speed, Held *et al.* [28] propose GOTURN tracker without online training, which learns a generic relationship between object motion and appearance from large number of videos offline and achieve 100+ FPS tracking speed. However, the accuracy is not satisfactory enough.

D. Technical Comparisons Among Similar Trackers

Although the idea of fusion tracking, learning, and detection together is not new in the literature, how to incorporate them properly to improve the tracking performance is still a challenging problem. Different from similar state-of-art trackers, the proposed approach uses the concept of event-triggering to

build a novel tracking framework, in which each module works independently and is only driven by its corresponding event. To achieve this, an occlusion and model drift detection model is constructed by measuring spatial and temporal tracking loss. Detailed technical comparisons are listed in Table I. According to the experimental results presented in Section V, the proposed tracker achieves significant improvement over the listed trackers in terms of both tracking accuracy and speed. Especially, even though LCT tracker discussed in the introduction has similar modules to the proposed tracker, as listed in Table I, the tracking accuracy of the proposed ETT is significantly higher than that of LCT while the tracking speed is almost the same.

TABLE I
Technical comparisons among similar state-of-art trackers.

	TLD [9]	LCT [11]	MUSTer [10]	ROT [12]	MDP [22]	ETT
occlusion identification	×	×	×	✓	×	✓
drift identification	×	✓	×	×	✓	✓
object detection	✓	✓	✓	×	✓	✓
re-detection on demand	×	✓	×	×	×	✓
weighted learning with occlusion	×	×	×	×	×	✓

III. EXISTING TECHNIQUES USED IN OUR PROPOSED TRACKER

In this section, we present some preliminaries about correlation filter tracking and online-SVM classifier.

A. Techniques in Correlation Filter-based Tracking

Recently, correlation filter-based trackers [1]–[3], [5], [10], [11] have enhanced the robustness and efficiency of visual tracking. Generally, a translation correlation filter F_t and scale correlation filter F_s are integrated together to perform translation and scale estimation respectively.

To perform translation estimation, a typical correlation filter F_t is trained on image patch $\mathbf{x}$ of $M \times N$ pixels, where all the cyclic shifts of $\mathbf{x}_{m,n}$ are training samples with $\mathbf{y}_{m,n}$ being the regression target for each $\mathbf{x}_{m,n}$, $(m, n) \in \{0, 1, \dots, M-1\} \times \{0, 1, \dots, N-1\}$. The goal is to find a function $g(\mathbf{z}) = \mathbf{w}^T \mathbf{z}$, where $\mathbf{w}$ is determined in such a way that the following objective is achieved:

$$\min_{\mathbf{w}} \sum_{m,n} |\langle \mathbf{w}, \phi(\mathbf{x}_{m,n}) \rangle - y_{m,n}|^2 + \lambda \|\mathbf{w}\|^2 \quad (1)$$

where ϕ denotes the mapping to a kernel space and $\lambda \geq 0$ is a regularization parameter.

Expressing the solution $\mathbf{w}$ as $\sum_{m,n} \alpha_{m,n} \phi(\mathbf{x}_{m,n})$ converts the optimization problem from variable $\mathbf{w}$ to α . The coefficient α can then be obtained based on the properties of circulant matrices:

$$\alpha = \mathcal{F}^{-1} \left(\frac{\mathcal{F}(\mathbf{y})}{\mathcal{F}(\langle \phi(\mathbf{x}), \phi(\mathbf{x}) \rangle) + \lambda} \right) \quad (2)$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ represent the Fourier transform and its inverse, respectively. From the properties of Fourier transform, we can compute the filtering response for all candidate patches by substituting (2) to $g(\mathbf{z})$

$$\hat{f}(\mathbf{z}) = \mathcal{F}^{-1}(\mathcal{F}(\langle \phi(\mathbf{z}), \phi(\mathbf{x}) \rangle) \odot \mathcal{F}(\alpha)) \quad (3)$$

where $\odot$ denotes the element-wise product and $\hat{g}(\mathbf{z})$ is an estimation of $g(\mathbf{z})$. The translation is estimated by finding the patch with the maximal value of $\hat{g}(\mathbf{z})$ through (3).

To handle the scale problem, a one-dimensional correlation filter F_s is trained on Q image patches, where pyramid around the estimated translation location is cropped from the image. Assume that the target size in the current frame is $W \times H$ and let P denote the account of scales $s \in \{\varepsilon^p | p = \lfloor -\frac{P-1}{2} \rfloor, \lfloor -\frac{P-3}{2} \rfloor, \dots, \lfloor \frac{P-1}{2} \rfloor\}$ where ε is a parameter representing the base value of scale changing. Then all Q image patches I_s with size $sW \times sH$ centered around the estimation location are resized to the template size for feature extraction. The optimal scale $\hat{s}$ is given as the highest correlation filter response of image patch I_s .

B. Techniques in Discriminative Online-SVM Classifier

We consider a linear classifier of the form $f(\mathbf{x}|\omega, b) = \omega^T \mathbf{x} + b$, where ω is the weight vector, b is the bias threshold. Suppose we have one previously trained model and a set of new training samples $\mathbf{x}_i \in \mathbb{R}^n$. The online learning algorithm trains the model at trial t is confined to use samples from past t trials, which can be regarded as solving the following optimization problem

$$\min_{\omega} \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^{t-1} \ell^\gamma(\omega; (\mathbf{x}_i, y_i)) \quad (4)$$

where $y_i \in \{1, -1\}$ is the binary label of $\mathbf{x}_i$, parameter C controls the trade-off between the complexity of ω and the cumulative hinge-loss of ω , ℓ^γ is the hinge-loss function [29].

Using the notion of duality, Eq (4) can be converted into its equivalent dual form as

$$\begin{aligned} \max_{\beta} \mathcal{D}_t(\beta) &= \sum_{i=1}^{t-1} \beta_i - \frac{1}{2} \sum_{i=1}^{t-1} \sum_{j=1}^{t-1} \beta_i \beta_j y_i y_j K_{ij} \\ 0 &\leq \alpha_i \leq C, \quad \sum_{i=1}^{t-1} \beta_i y_i = 0, \end{aligned} \quad (5)$$

where $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ is a symmetric positive definite function, such as Gaussian function.

Therefore, the online-SVM algorithm can be treated as an incremental solver of dual problem [29], where at the end of trial t the algorithm maximizes the dual function confined to the first $t-1$ observed variables. Finally, the optimal parameter ω could be easily obtained by using existing convex optimization tools, such as LIBSVM [30].

Algorithm 1 Event-triggered tracker

input: initial bounding box r_0 , image sequence $\{I_t\}_0^T$

output: estimated bounding boxes $\{B_t^e\}_1^T$

Initialize correlation filter and online-SVM classifier by using B_0^e and I_0

for $t = 1 : T$ **do**

$B_c \leftarrow$ correlation filter prediction bounding box

$\mathcal{L}_{\mathbf{x}^\circ, \mathbf{p}^\circ}^t \leftarrow \delta \mathcal{L}_{\mathbf{x}^\circ}^S + (1 - \delta) \mathcal{L}_{\mathbf{p}^\circ}^T$

// Triggering each events

if $\mathcal{L}_{\mathbf{x}^\circ, \mathbf{p}^\circ}^t > T_{fail}$

Do object re-detection by finding $B_r \leftarrow \max_{h_i} \omega^T \Phi(B_i) + b$

Reinitialize correlation filter by patch (I_t, B_r)

$B_t^e \leftarrow B_r$ & **Continue** tracking next frame

if $T_{occ} < \mathcal{L}_{\mathbf{x}^\circ, \mathbf{p}^\circ}^t < T_{fail}$

$\xi_o \leftarrow \xi_n \frac{T_{fail} - \mathcal{L}_{\mathbf{x}^\circ, \mathbf{p}^\circ}^t}{T_{fail} - T_{occ}}$

if $\mathcal{L}_{\mathbf{x}^\circ, \mathbf{p}^\circ}^t < T_{occ}$

$\xi_o \leftarrow \xi_n$

Do re-sampling for update detector model

if $\mathcal{L}_{\mathbf{x}^\circ, \mathbf{p}^\circ}^t < T_{ap}$ **and** $m_{rs} > \epsilon$

Do detecting model update

Do correlation filter update

$B_t^e \leftarrow B_c$

end for

IV. THE PROPOSED TECHNIQUES

In this section, the details of newly proposed techniques including occlusion and model drift detection and event-triggering conditions are given with reference to Figure 1.

A. Occlusion and Model Drift Detection

One of major issues in visual tracking is to handle model drift and also occlusion, as they are prone to result in tracking failure. Factors leading to model drift include illumination variation, fast movement, appearance change and occlusion. It should be noted that occlusion can be considered as a separate issue for its generality and challenge in visual tracking. The drift problem could also be alleviated by adjusting the learning strategy of short-term tracking model when occlusion occurs.

To better assess the short-term tracking process, two criteria are defined and then fused to verify current tracking state:

Criterion 1 (Spatial locality-constraint). *For any accurate tracking frame, a patch which is closer to the center of the prediction should be more similar to the target, and vice versa. Otherwise, this criterion cannot be met if the present tracking is incorrect or in the presence of occlusion or drift.*

Denote $\mathbb{E}_S(\cdot, \cdot)$ as a metric that defines the similarity between two patches, c_{gt} is the center of ground truth patch, c_i and c_j are

the centers of candidate patches. According to Criterion 1, an accurate tracking frame should satisfy the following condition.

$$\mathbb{E}_S(c_{gt}, c_i) < \mathbb{E}_S(c_{gt}, c_j) \text{ if } \|c_{gt} - c_i\|_2 < \|c_{gt} - c_j\|_2 \quad (6)$$

Criterion 2 (Temporal continuity). *The cumulative estimation loss for short-term tracking within a recent temporal window should be close to zero if the predictions given by short-term tracker are accurate or without drift, and vice versa.*

Denote $\mathbb{E}_T(\cdot, \cdot)$ as a metric for measuring estimation loss. $P(\cdot)$ and $P'(\cdot)$ are two prediction models, $P_{gt}(\cdot)$ denotes the ground truth. From Criterion 2, an accurate tracking in a predefined time window should obey the following condition if $P(\cdot)$ is more accurate than $P'(\cdot)$.

$$\sum_{i=0}^{\Delta} \mathbb{E}_T(P(t-i), P_{gt}(t-i)) < \sum_{i=0}^{\Delta} \mathbb{E}_T(P'(t-i), P_{gt}(t-i)) \quad (7)$$

where t is the number of tracking frame. Note that the cumulation loss is computed in a time window to achieve a robust loss estimation.

Based on Criteria 1 and 2, two tracking losses are proposed and fused to evaluate the short-term tracking performance.

1) *Spatial Loss Evaluation*: Following Criterion 1, a spatial loss evaluation is designed. Denote all the possible candidate patches within a square neighborhood of the t -th prediction patch $x_{p,t}^\circ$ as $\mathbf{x}^\circ = \{I_t, u_i, v_i, w_{p,t}, h_{p,t}\}_{i=1}^{N_s}$, where I_t is the image of the t -th frame, (u_i, v_i) denotes the position, $(w_{p,t}, h_{p,t})$ represents the scale of current prediction $x_{p,t}^\circ$, N_s is the number of patches. A real-valued linear function $f(\Phi(x_i^\circ)|\omega, b) = \omega^T \Phi(x_i^\circ) + b$ with details given in III-B is employed to estimate the confidence score for each patch x_i° , where Φ is a mapping from the original input space to the feature space. Meanwhile, the label for x_i° is produced by calculating the overlap rate with $x_{p,t}^\circ$.

$$L^\circ(x_i^\circ|x_{p,t}^\circ) = \frac{\text{area}(ROI_{x_i^\circ} \cap ROI_{x_{p,t}^\circ})}{\text{area}(ROI_{x_i^\circ} \cup ROI_{x_{p,t}^\circ})} \quad (8)$$

In this paper, we use Kullback-Leibler divergence [31] to assess spatial loss defined as:

$$\mathcal{L}_{\mathbf{x}^\circ}^S = \sum_{x_i^\circ \in \mathbf{x}^\circ} \bar{f}(\Phi(x_i^\circ)|\omega, b) \log \frac{\bar{f}(\Phi(x_i^\circ)|\omega, b)}{\bar{L}^\circ(x_i^\circ)} \quad (9)$$

where $\bar{f}$ and $\bar{L}^\circ$ are the normalized form.

2) *Temporal Loss Evaluation*: Such an evaluation is designed according to Criterion 2. The main objective of the temporal loss function is to evaluate the accuracy of the short-term tracker in a recent time period. Let the desired confidence score be L^T . Denote the estimated patches of recent Δ predictions as $\mathbf{p}^\circ = \{I_i, u_i, v_i, w_i, h_i\}_{i=t-\Delta}^t$, where Δ is the size of the temporal window. The temporal loss function is defined as:

$$\mathcal{L}_{\mathbf{p}^\circ}^T = \frac{1}{\Delta + 1} \sum_{i=t-\Delta}^t \varpi_{t,i} \|f(\Phi(p_i^\circ)|\omega, b) - L^T\|_2^2 \quad (10)$$

where $\|\cdot\|_2$ is 2-norm, $\varpi_{t,i}$ denotes the weight for each past prediction, such as forgetting curve $e^{-\frac{t}{\rho}}$ and uniform distribution. Set L^T for each prediction to +1 since the confidence $f(\Phi(x_i^\circ)|\omega, b)$ should be close to +1 if current prediction is accurate.

Fig. 2. Illustration for drift detection and restoration. The bounding boxes colored in green, red solid, red dash and yellow denote ground truth, ETT prediction, the prediction without re-detection and corrected results after detecting tracking failure, respectively.

Finally, the tracking loss at the t -th frame is the fusion of spatial loss and temporal loss.

$$\mathcal{L}_{\mathbf{x}^\circ, \mathbf{p}^\circ}^t = \delta \mathcal{L}_{\mathbf{x}^\circ}^S + (1 - \delta) \mathcal{L}_{\mathbf{p}^\circ}^T \quad (11)$$

where $\delta \in [0, 1]$, which is a weighting factor to balance the spatial loss and temporal loss.

A smaller tracking loss represents better tracking performance. Two thresholds T_{fail} and T_{occ} are initialized to classify the tracking state into normal tracking, partial occlusion, and drift/heavy occlusion. The tracking state is set to drift if $\mathcal{L}_{\mathbf{x}^\circ, \mathbf{p}^\circ}^t > T_{fail}$, or partial occlusion if $T_{fail} \geq \mathcal{L}_{\mathbf{x}^\circ, \mathbf{p}^\circ}^t > T_{occ}$, otherwise set to normal tracking. However, a fixed threshold may not be an appropriate way for various video sequences. In this paper, a threshold learning approach is proposed to update T_{fail} and T_{occ} adaptively by considering the recent tracking losses $\mathcal{L}_{\mathbf{x}^\circ, \mathbf{p}^\circ}^t = \{\mathcal{L}_{\mathbf{x}^\circ, \mathbf{p}^\circ}^i\}_{i=t-\Delta}^t$ in relation to the current threshold. Let θ_d and θ_o denote the learning rate for T_{fail} and T_{occ} respectively. The adaptive thresholds are defined as

$$T = T + \theta \left(\frac{1}{\Delta + 1} \sum \mathcal{L}_{\mathbf{x}^\circ, \mathbf{p}^\circ}^t - T \right) \quad (12)$$

where $T = \{T_{fail}, T_{occ}\}$ and $\theta = \{\theta_d, \theta_o\}$. An upper and a lower bound are set for T to avoid adaptive threshold becoming too large or too small. Figure 2 illustrates the process of drift detection and restoration. The black line denotes the tracking loss for each frame, and the blue line is the tracking drift threshold, which is adaptively changed according to (12).

B. Event-triggered Decision Model

As previously mentioned, the tracker consists of multiple modules, each of which is intended to conduct a subtask of the entire visual tracking. An event for each module is associated with a set of conditions. When those certain conditions for an event are met, the event becomes active and triggers the corresponding subtask to be executed. The details of the proposed events are summarized in Table II.

Let $\mathbf{E}$ denote the event set, i.e. $\mathbf{E} = \{\mathbf{E}_{ctmu1}, \mathbf{E}_{ctmu2}, \mathbf{E}_{rdm}, \mathbf{E}_{hor}, \mathbf{E}_{dmu}, \mathbf{E}_{nt}\}$. Each event in the set is independent of the

TABLE II
The feature representation for the defined events.

Event	Notation	Function Description
Correlation Tracking Model Updating 1 (CTMU1)	$\mathbf{E}_{ctmu1}$	Updating the correlation tracking model when no model drift or occlusion happens
Correlation Tracking Model Updating 2 (CTMU2)	$\mathbf{E}_{ctmu2}$	Updating the correlation tracking model when partial occlusion happens
Heuristic Object Re-detection (HOR)	$\mathbf{E}_{hor}$	Re-detecting the target when model drift happens
Re-sampling for Detector Model (RDM)	$\mathbf{E}_{rdm}$	Extracting the new positive and negative samples based on current frame and pushing them to the sampling-pool when the prediction is accurate
Detector Model Updating (DMU)	$\mathbf{E}_{dmu}$	Updating the detector model when the model is not suitable for current appearance
Normal Tracking (NT)	$\mathbf{E}_{nt}$	Continuing the short-term tracking when no drift is detected

Fig. 3. Illustration of the event-triggered decision tree. The event-triggered decision model will produce one or multiple events based on the input from occlusion and drift identification model, which provides the information of short-term tracking state evaluation.

others and has two states, i.e active or inactive, represented by 1 or 0, respectively, to indicate the executing status of its corresponding subtask. If all the conditions of an event are met, then its state turns to active. For instance, the state of $\mathbf{E}_{hor}$ being active indicates that model drift is detected, then the re-detection module will be triggered to relocate the target, otherwise re-detection module is in inactive (sleeping) state. During the tracking process, the aforementioned six events are verified sequentially and the state for each event is set accordingly.

It is essential to cooperatively integrate these events such that the tracker is able to carry out the entire task efficiently and robustly. In this paper, we use the information collected from occlusion and drift diagnosis model, as detailed in Section IV-A, and a decision tree to build an event-triggered decision model, as shown in Figure 3.

1) *Correlation Tracking Model Updating*: The correlation tracking state could be classified into normal tracking, tracking with partial occlusion and tracking with model drift or heavy occlusion. Generally, model updating for tracker and detector should be carried out when the first two situations are met but not for model drift or heavy occlusion, since noisy samples will cause model pollution. However, long-term partial occlusion also has a great impact on the model, thus using a learning rate smaller than that in normal tracking is necessary. According to

the discussions in Section IV-A, the corresponding events for updating short-term tracker on the situation of normal tracking and partial occlusion are proposed respectively as

$$\begin{cases} \mathbf{E}_{ctmu1} = \{\mathcal{L}_{\mathbf{x}^\circ, \mathbf{p}^\circ}^t < T_{occ}\} \\ \mathbf{E}_{ctmu2} = \{T_{occ} \leq \mathcal{L}_{\mathbf{x}^\circ, \mathbf{p}^\circ}^t < T_{fail}\}. \end{cases} \quad (13)$$

Based on the aforementioned observations, the learning rate for correlation tracker is computed as:

$$\xi_o = \begin{cases} \xi_n, & \text{if } \mathbf{E}_{ctmu1} = 1 \\ \xi_n \frac{T_{fail} - \mathcal{L}_{\mathbf{x}^\circ, \mathbf{p}^\circ}^t}{T_{fail} - T_{occ}}, & \text{if } \mathbf{E}_{ctmu2} = 1 \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

where ξ_n is the learning rate for normal tracking. The learning rate on the situation of partial occlusion decreases linearly from ξ_n according to occlusion degree.

Therefore, the translation filter F_t and scale filter F_s in Section III-A could be updated with the learning rate ξ_o as

$$\begin{cases} \hat{\mathbf{x}}^t = (1 - \xi_o)\hat{\mathbf{x}}^{t-1} + \xi_o \mathbf{x}^t \\ \hat{\alpha}^t = (1 - \xi_o)\hat{\alpha}^{t-1} + \xi_o \alpha^t \end{cases} \quad (15)$$

where $\hat{\mathbf{x}}^t$ and $\hat{\alpha}^t$ are the learned object appearance and parameter of correlation filter, respectively. They are considered as the respective estimation of $\mathbf{x}^t$ and α^t , and thus used to determine $\hat{g}(\mathbf{z})$ in (3).

2) *Heuristic Object Re-detection*: The object re-detection module is triggered when tracking failure is detected. Thus the event $\mathbf{E}_{hor}$ is proposed as follows:

$$\mathbf{E}_{hor} = \{\mathcal{L}_{\mathbf{x}^\circ, \mathbf{p}^\circ}^t > T_{fail}\} \quad (16)$$

In this case, most existing short-term trackers cannot recover from drift since the model is contaminated due to inappropriate updating on noisy samples. To overcome this problem, a new discriminative appearance model should be used to relocate the target and reinitialize the currently polluted short-term tracker.

In the proposed tracking algorithm, we implement a discriminative online-SVM classifier detailed in Section III-B as our detector. Given a frame I , the state of a sample is represented by a bounding box $\mathbf{B} = \{I_t, u_i, v_i, w_i, h_i\}_{i=1}^{N_b}$. The feature extracting from sample B_i is denoted as $\mathbf{h}_i = \Phi(B_i)$. Then the detection task is converted to the following problem:

$$\max_{\mathbf{h}_i} f(\mathbf{h}_i | \omega, b) = \max_{\mathbf{h}_i} \omega^T \Phi(B_i) + b \quad (17)$$

where ω and b are the optimized parameters of SVM model.

As the number of bounding boxes N_b to be evaluated is large for the holistic frame, the classification process requires a huge amount of computation. However, the detection process could be accelerated based on the past tracking performance, which provides important information about the area where the target possibly locates. Let $\mathcal{C}(b_c, r)$ denote the circular region with center point b_c and radius r . Suppose that r_b is a predefined basic radius. Then according to mean value of past tracking loss, r could be adaptively adjusted as

$$r = r_b \left(1 + \frac{1}{\Delta + 1} \sum_{i=t-\Delta}^t \mathcal{L}_{\mathbf{x}^\circ, \mathbf{p}^\circ}^i\right) \quad (18)$$

because the bigger the mean tracking loss indicates, the higher the possibility that the target is lost or far away from current

prediction. By considering the *priori* tracking information, the detector could always find an optimal searching region without sacrificing accuracy.

3) *Detector Model Updating*: A robust tracking model normally needs to be updated continuously based on current prediction to capture the appearance variation of the target. However, this semi-supervised learning has its drawbacks. Firstly, it is sensitive to new samples. The noisy or mislabeled samples will cause catastrophic results to the original model. Secondly, due to continuous accumulation of new samples, the sampling-pool would become increasingly larger, which significantly slows down the on-line learning process.

To overcome the above problems, the following sampling-pool $\mathcal{T}$ is constructed to robustly update SVM model.

$$\mathcal{T} = \mathcal{T}_{sup} \cup \mathcal{T}_{conf} \cup \mathcal{T}_{rs} \quad (19)$$

where $\mathcal{T}_{sup}$ denotes support samples from previous SVM model, $\mathcal{T}_{conf}$ is the set of samples with high confidence of being positive or negative, which is obtained by collecting the n_{sup} farthest samples from the model margin. The primary objective of $\mathcal{T}_{conf}$ is to keep the model robust to noisy or mislabeled samples. $\mathcal{T}_{rs}$ is the set of new samples sampled between previous training stamp and current training stamp. The maximum capacities of $\mathcal{T}_{conf}$ and $\mathcal{T}_{rs}$ are set as M_{conf} and M_{rs} respectively. When the maximum capacity is reached, some samples need to be removed randomly. Set $\mathcal{T}_{rs}$ will be reset to empty once the SVM model is retrained. Figure 4 illustrates the principle of sampling-pool.

It is unnecessary to update the detector model as frequently as short-term tracking model since the detector model is not sensitive to small variations of appearance among consecutive frames, while the retraining process is time-consuming. Generally, the updating should be executed when the model is not suitable for the current appearance, which may be caused by, for example, large appearance variation. The reasons are: 1) samples with small appearance change have a high probability of locating outside the margin or overlapping with existing support vectors based on the Karush-Kuhn-Tucker (KKT) conditions [32], which have little impact on SVM model. 2) the samples with high appearance variation are likely to reside around the margin of SVM model, which would have a remarkable influence on the current model and make it more generalized. With the above observations, and letting $m_{rs} = m_{rs+} + m_{rs-}$, which is the current size of $\mathcal{T}_{rs}$, an event $\mathbf{E}_{dmu}$ is now proposed to activate the updating of SVM model as follows:

$$\mathbf{E}_{dmu} = \{\mathcal{L}_{x^o, p^o}^t < T_{ap} \cap m_{rs} > \epsilon\} \quad (20)$$

where ϵ and T_{ap} are predefined constants. T_{ap} is a threshold for huge appearance changing.

4) *Re-sampling for Detector Model*: Re-sampling happens when triggered by the event defined in Eq (21). Then the event $\mathbf{E}_{rdm}$ is proposed as

$$\mathbf{E}_{rdm} = \{\mathcal{L}_{x^o, p^o}^t < T_{occ}\} \quad (21)$$

which means that the current prediction is accurate. In this case, the tracker will sample n_{rs+} patches around prediction as positive samples and n_{rs-} patches far away from prediction as negative samples, which are then added to $\mathcal{T}_{rs}$. When $\mathcal{T}_{rs} \geq M_{rs}$ is full, some redundant samples will be removed from the sampling-pool $\mathcal{T}_{rs}$.

Fig. 4. Illustration of the representation: (a) shows the composition of sampling-pool, which is divided into three portions, namely, support samples, high confidence samples and re-sampled samples; and (b) shows the updating methods for the first two parts in (a).

5) *Normal Tracking*: The event representing normal tracking is defined as

$$\mathbf{E}_{nt} = \mathbf{E}_{ctmu1} \cap \mathbf{E}_{ctmu2} = \{\mathcal{L}_{x^o, p^o}^t < T_{fail}\} \quad (22)$$

It also indicates that there is no model drift detected. In this case, the short-term tracker could continuously perform tracking task even with partial occlusion, and the re-detection module will not be activated for the disturbance from environment is under the tolerance of short-term tracker. In this case, the prediction given by short-term tracker is still considered reliable. Overall, the proposed method is illustrated in Algorithm 1.

V. EXPERIMENTS

A. Implementation Details

Two different types of features are applied to the proposed tracking algorithm. Specifically, HOG features with 31 bins and 4x4 cell size are used in the short-term tracker, while 300 Haar-like features with 6 different types and 50 bounding boxes are adopted to train and test the online-SVM classifier. The parameters for the proposed ETT tracker are specified as follows: The parameters for correlation filter based short-term tracking is the same as in [5]. The number of past predictions Δ is used to detect occlusion and tracking failure/drift, which should neither be too small to make drift detection sensitive, nor too big to make drift detection unresponsive or dullness. Empirically, $\Delta = 20$ will provide a quite satisfactory performance. The initialized occlusion threshold T_{occ} and failure/drift threshold T_{fail} are set as 0.3 and 0.6 respectively. The lower bounds for T_{occ} and T_{fail} are set as 0.2 and 0.4, respectively. The upper bounds are 0.4 and 0.8, respectively. The learning rate θ_d and θ_o for T_{fail} and T_{occ} are set to 0.01 and 0.015 respectively. The threshold for huge appearance changing T_{ap} is set to 0.25. The value of δ is set to 0.45. Empirically, there is little influence for tracking accuracy when δ is set between 0.3 to 0.7. In all the

Fig. 5. Quantitative results on the benchmark datasets. The scores in the legends indicate the average Area-Under-Curve value for precision and success plots respectively.

experimental tests with different schemes, these parameters are set to be the same for the whole testing sequences.

TABLE III

Comparisons among baseline trackers with the best results displayed in red.

	TB-50 [15]		TB-100 [15]		Mean FPS
	DP (%)	OP (%)	DP (%)	OP (%)	
DSST [5]	57.7	46.4	63.0	52.0	37.7
$ETT-RD$	59.4	49.1	66.0	55.2	27.0
$ETT-FD$	66.9	54.5	73.6	61.1	6.5
$ETT-DU$	67.8	55.4	73.7	61.4	21.0
ETT	71.3	59.0	76.2	63.7	18.1

B. Evaluation on CVPR2013 [33] and TB-100 [15] Dataset

The CVPR2013 [33] contains 50 videos. The TB-100 [15] is expanded from CVPR2013 [33] which contains 100 videos. In order to facilitate an in-depth analysis, the authors in [15] select 50 difficult and representative sequences from the whole dataset. This subset is denoted as TB-50 [15]. Note that the 50 sequences in TB-50 are different from those in CVPR2013 [33].

a) *Components Analysis*: To evaluate the contribution made by each module, we test following variations of the proposed method with a combination of different modules. a) $ETT-BASE$: the baseline tracker of the proposed method. In our method, we implement DSST [5] as our baseline tracker. b) $ETT-RD$: the ETT tracker without target re-detection module. c) $ETT-FD$: the ETT tracker without occlusion and failure detection module which means detection will occur at each frame. d) $ETT-DU$: the ETT tracker without online discriminate learning for detector module, the detector model

is trained by using samples collected from first 5 frames and then it is fixed in the following frames. e) ETT : the proposed overall event-triggered tracker.

As shown in Table III, it can be concluded that 1) weighted learning for correlation filter tracker has provided certain ability to address occlusion when comparing $ETT-RD$ with baseline tracker. 2) Occlusion and drift detection is useful when comparing $ETT-FD$ with ETT . Due to the occlusion and drift detection module in ETT , the tracker could alleviate the influence from noisy samples and improve the accuracy. On the other hand, it greatly increases the tracking speed by re-detecting the target only when needed. 3) Comparing $ETT-DU$ with baseline tracker, it can be concluded that re-detection is an effective way to restore tracking failure. $ETT-DU$ has a weak detector since it only trains a detection model by the samples collected from first 5 frames, even though $ETT-DU$ improves more than 9 percent compared with baseline tracker. 4) The online discriminate learning is helpful when comparing ETT with $ETT-DU$, since it could improve the adaptability of detector to handle various environment changing. With all the modules, the proposed tracker improves around 11% than the baseline tracker in terms of overlap precision.

b) *Quantitative Evaluation*: The proposed ETT approach is compared with eleven state-of-the-art trackers (MUSTer [10], LCT [11], LCT-Deep [19], SRDCF [17], Staple [3], MEEM [16], DSST [5], TLD [9], ROT [12], KCF [1] and STRUCK [6]) on CVPR2013 dataset [33] and its extended version TB-100 dataset [15]. The precision and success plots¹ over these datasets are presented in Figure 5.

According to Figure 5, it is observed that the proposed method achieves a similar accuracy on CVPR2013 [33] compared with LCT-Deep [19] and MUSTer [10], while surpasses the state-of-the-art trackers both in accuracy and precision by a large margin on TB-100 [15]. Especially on more challenging dataset TB-50 [15], the proposed method also achieves favorable performances when compared with other trackers. According to Figure 5 (a) and (c), due to more challenging sequences added to the CVPR2013 [33] dataset, including great deformation, long-term occlusion *et al.*, the performance of all other trackers are degraded while the proposed method almost maintains the same performance. This is owing to that the proposed failure detection and event-triggered target re-detection can promptly and effectively handle these challenging attributes. As shown in Figure 5 (b) and (c), the proposed method outperforms the second best tracker (LCT-Deep and SRDCF) by almost 4 percent in terms of overlap success rate and distance precision.

Actually, qualitative comparisons are also conducted. But due to page limit, the comparison results are not included this paper. These results demonstrate that our tracker is able to locate the targets in these complicated scenarios more precisely than the other state-of-the-art trackers. Overall, the proposed ETT tracker outperforms the state-of-the-art trackers in terms of overlap success and distance precision.

c) *Comparisons on different attributes on TB-100*: To thoroughly evaluate the robustness of the proposed ETT tracker in various scenes, we present tracking accuracy in terms of 8

¹The average Area-Under-Curve score is different with the score at a specific threshold, i.e. the distance precision score of MEEM in CVPR2013 at the threshold of 20 pixels is 0.83 while the average AUC score is 0.743.

TABLE IV

A time consumption comparison. The mean overlap precision (OP) (%), distance precision (DP) (%) and mean FPS over all the 100 videos in the TB-100 [15] are presented. The two best results are displayed in red and blue, respectively.

	ETT	MUSTer	LCT-Deep	SRDCF	LCT	MEEM	Staple	DSST	KCF	STRUCK	TLD	ROT
Mean OP	63.7	58.9	59.4	60.0	57.4	53.9	58.1	52.0	49.1	46.4	40.9	48.7
Mean DP	76.2	72.1	73.6	71.8	69.9	71.1	71.3	63.2	64.6	59.6	52.5	60.0
Mean FPS	18.1	5.0	17.9	6.6	17.1	13.5	17.6	37.7	245	10.5	25.3	28.6

Fig. 6. Quantitative results on TB-100 [33] for 8 challenging attributes: illumination variation, scale variation, occlusion, deformation, motion blur, out-of-plane rotation, out-of-view, background clutters.

challenging attributes on TB-100 [15] in Figure 6. As observed from the figure, the proposed tracker ETT outperforms other methods by a huge margin in all the attributes, particularly in handling occlusion, deformation, scale variation, out-of-plane rotation and out-of-view, which can be attributed to the effectiveness of event-triggered detection approach in relocating target when tracking drifts occur. It also demonstrates that the proposed event based triggering mechanism can accurately recognize tracking failure effectively. Note that the trackers MUSTer, LCT, LCT-Deep and TLD also include a detection module, however, they show their limitations in handling these challenging situations.

d) *Comparisons on tracking speed:* With the configuration in Section V-A, we compare the speed of the 11 trackers on a platform with Intel Xeon E5-1630 3.70 GHz CPU and 16.0 GB RAM. The results for all these trackers are listed in Table IV, based on which the following observations are made. 1) It is seen that ETT tracker not only achieves the highest tracking precision but also operates in real-time. 2) Compared to the trackers LCT and MUSTer, which also adopt re-detection modules, ETT tracker improves significantly both on distance precision and overlap precision, and in the meantime, its tracking speed is faster than that of LCT and MUSTer. 3) Although trackers KCF and DSST give faster tracking speed, their tracking accuracies are unsatisfactory. On the other hand, our ETT tracker improves significantly on tracking accuracy, while with a similar tracking speed when compared to MEEM, Staple, LCT, and SRDCF.

C. Evaluation on VOT16 [27] Dataset

In addition to TB-100, we also evaluate the proposed method on VOT16 [27], which contains 60 challenging real-life videos.

We compare our tracker with the following state-of-the-art methods: SCT4 [34], SODLT [35], DNT [36], DSST2014 [5], KCF2014 [1], ANT [37], ASMS [38], TricTRACK [39], MAD [40], SiamAN [41], SRDCF [17], TGPR [42], Staple [3] and DeepSRDCF [17].

As shown in Figure 7, the proposed tracker ranks first in terms of accuracy and third in terms of robustness. Among these competitive trackers, the deep learning based ones (SODLT [35], DNT [36], SiamAN [41] and DeepSRDCF [17]) generally achieve a higher rank in robustness. However, the proposed tracker achieves a good trade-off between robustness and accuracy which makes our methods competitive to the other trackers. Note that, the deep learning based trackers usually have a low tracking speed [43] and require additional hardware support of GPU. Our proposed tracker runs on CPU only and can achieve a speed of 18 FPS, which makes it suitable for real-time applications.

D. Discussion on Failure Cases

The proposed tracker may not perform well for some extremely challenging cases such as the *hand*, *fish1* in VOT16 [27] and the *Diving*, *Jump* in TB-100 [15]. For the former sequences, the target is quite similar to interference of surroundings, resulting in that the ETT can hardly extract useful spatial and temporal loss information to verify current tracking performance. In such an environment, the proposed tracker could not effectively identify the drift cases since Criteria 1 and 2 are hardly satisfied. For the latter sequences, the ETT fails to restore the tracker from tracking drift due to the limitation of features used to describe the target and candidates. In fact, the aforementioned sequences are so challenging that most of the

Fig. 7. Accuracy-robustness ranking plot for the state-of-the-art comparison.

trackers cannot perform well. On the other hand, the comparison results on these two large datasets with various objects demonstrate that the proposed tracking algorithm is able to detect occlusion and drift accurately, which promptly triggers to address the tracking failure by re-detecting the target. As such, higher tracking accuracy is achieved while the tracking speed is also improved. In conclusion, based on the experimental results, the proposed ETT approach is proven to be sufficiently effective and efficient to handle various environmental challenges.

VI. CONCLUSIONS

In this paper, we present a simple but effective tracking framework by proposing event-triggered tracking algorithms with occlusion and drift detection. The overall tracker is decomposed into several independent modules involving different subtasks. An event-triggered decision model is proposed to coordinate those modules in various scenarios. Moreover, a novel occlusion and drift detection algorithm is proposed in this paper to tackle the general yet challenging drift and occlusion problems. Our tracking algorithm outperforms the state-of-the-art trackers in terms of both tracking speed and accuracy, in TB-100 [15] and VOT16 [27] benchmarks.

REFERENCES

- [1] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, 2015.
- [2] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Comput. Vision and Pattern Recognition*, pp. 2544–2550. IEEE, 2010.
- [3] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. Torr, "Staple: Complementary learners for real-time tracking," *arXiv preprint arXiv:1512.01355*, 2015.
- [4] X. Zhang, W. Hu, S. Chen, and S. Maybank, "Graph-embedding-based learning for robust object tracking," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 2, pp. 1072–1084, 2014.
- [5] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *British Mach. Vision Conf.*, 2014.
- [6] S. Hare, A. Saffari, and P. H. Torr, "Struck: Structured output tracking with kernels," in *Int. Conf. Comput. Vision*, pp. 263–270. IEEE, 2011.
- [7] H. Song, Y. Zheng, and K. Zhang, "Robust visual tracking via self-similarity learning," *Electronics Letters*, vol. 53, no. 1, pp. 20–22, 2016.
- [8] K. Zhang, Q. Liu, J. Yang, and M.-H. Yang, "Visual tracking via boolean map representations," *Pattern Recognition*, vol. 81, pp. 147 – 160, 2018.
- [9] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.
- [10] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking," in *Comput. Vision and Pattern Recognition*, pp. 749–758, 2015.
- [11] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Long-term correlation tracking," in *Comput. Vision and Pattern Recognition*, pp. 5388–5396, 2015.
- [12] X. Dong, J. Shen, D. Yu, W. Wang, J. Liu, and H. Huang, "Occlusion-aware real-time object tracking," *IEEE Transactions on Multimedia*, vol. 19, no. 4, pp. 763–771, 2017.
- [13] X. Wang and M. D. Lemmon, "Event-triggering in distributed networked control systems," *IEEE Transactions on Automatic Control*, vol. 56, no. 3, pp. 586–601, 2011.
- [14] L. Xing, C. Wen, Z. Liu, H. Su, and J. Cai, "Event-triggered adaptive control for a class of uncertain nonlinear systems," *IEEE Transactions on Automatic Control*, 2016.
- [15] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, 2015.
- [16] J. Zhang, S. Ma, and S. Sclaroff, "Meem: robust tracking via multiple experts using entropy minimization," in *European Conf. on Comput. Vision*, pp. 188–203. Springer, 2014.
- [17] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Int. Conf. Comput. Vision*, pp. 4310–4318, 2015.
- [18] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1442–1468, 2014.
- [19] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Learning adaptive correlation filters with long-term and short-term memory for object tracking," *International Journal of Computer Vision (IJCV)*, 2018.
- [20] X. Yang, M. Wang, L. Zhang, F. Sun, R. Hong, and M. Qi, "An efficient tracking system by orthogonalized templates," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 5, pp. 3187–3197, 2016.
- [21] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time object tracking via online discriminative feature selection," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 4664–4677, 2013.
- [22] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: Online multi-object tracking by decision making," in *Int. Conf. Comput. Vision*, pp. 4705–4713, 2015.
- [23] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *European conference on computer vision*, pp. 702–715. Springer, 2012.
- [24] R. Tao, E. Gavves, and A. W. Smeulders, "Siamese instance search for tracking," in *Comput. Vision and Pattern Recognition*, pp. 1420–1429. IEEE, 2016.
- [25] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in *Int. Conf. Comput. Vision*, 2015.
- [26] K. Zhang, Q. Liu, Y. Wu, and M.-H. Yang, "Robust visual tracking via convolutional networks without training," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1779–1792, 2016.
- [27] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Čehovin, T. Vojir, G. Häger, A. Lukežič, and G. Fernandez, "The visual object tracking vot2016 challenge results," Springer, Oct. 2016.
- [28] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 fps with deep regression networks," in *European Conf. on Comput. Vision*, pp. 749–765. Springer, 2016.
- [29] S. Shalev-Shwartz and Y. Singer, "Online learning meets optimization in the dual," in *Int. Conf. on Comput. Learn. Theory*, pp. 423–437. Springer, 2006.
- [30] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 27:1–27:27, 2011.
- [31] D. Johnson and S. Sinanovic, "Symmetrizing the kullback-leibler distance," *IEEE Transactions on Information Theory*, 2001.
- [32] B. Schölkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.
- [33] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Comput. Vision and Pattern Recognition*, pp. 2411–2418, 2013.
- [34] J. Choi, H. J. Chang, J. Jeong, Y. Demiris, and J. Y. Choi, "Visual tracking using attention-modulated disintegration and integration," in *Comput. Vision and Pattern Recognition*, Jun. 2016.
- [35] N. Wang, S. Li, A. Gupta, and D.-Y. Yeung, "Transferring rich feature hierarchies for robust visual tracking," *arXiv preprint arXiv:1501.04587*, 2015.
- [36] Z. Chi, H. Li, H. Lu, and M.-H. Yang, "Dual deep network for visual tracking," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 2005–2015, 2017.

- [37] L. Čehovin, A. Leonardis, and M. Kristan, "Robust visual tracking using template anchors," in *Applications of Computer Vision (WACV)*, 2016 *IEEE Winter Conference on*, pp. 1–8. IEEE, 2016.
- [38] T. Vojir, J. Nuskova, and J. Matas, "Robust scale-adaptive mean-shift for tracking," *Pattern Recognition Letters*, vol. 49, pp. 250–258, 2014.
- [39] X. Wang, M. Valstar, B. Martinez, M. Haris Khan, and T. Pridmore, "Tric-track: Tracking by regression with incrementally learned cascades," in *Int. Conf. Comput. Vision*, pp. 4337–4345, 2015.
- [40] S. Becker, S. B. Krah, W. Hübner, and M. Arens, "Mad for visual tracker fusion," in *Optics and Photonics for Counterterrorism, Crime Fighting, and Defence XII*, vol. 9995, p. 99950L, 2016.
- [41] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *European conference on computer vision*, pp. 850–865. Springer, 2016.
- [42] J. Gao, H. Ling, W. Hu, and J. Xing, "Transfer learning based visual tracking with gaussian processes regression," in *European Conf. on Comput. Vision*, pp. 188–203. Springer, 2014.
- [43] "Otb-100 benchmark results." [Online]. Available: https://github.com/foolwood/benchmark_results

Mingyang Guan received the B.S. and M.S. degree from Xi'an Jiaotong University, Xi'an, China, in 2010 and 2013, respectively. He is currently working toward the Ph.D. degree in the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore.

His current research interests include robots, machine learning, and navigation systems.

Mao Shan (M'12) received his B.S. degree in electrical engineering in 2006 from Shaanxi University of Science and Technology, Xi'an, China, and the M.S. and Ph.D. degrees from The University of Sydney, Australia, in 2009 and 2014, respectively. He was a Research Associate at The University of Sydney, Australia, from 2014 to 2016, and a Research Fellow at Nanyang Technological University, Singapore, from 2016 to 2017. Currently, he is a Research Fellow at the Australian Centre for Field Robotics, The University of Sydney, Australia.

His research interests include autonomous systems, localization and tracking algorithms and applications.

His research interests include autonomous systems, localization and tracking algorithms and applications.

Changyun Wen (F'10) received B.Eng. degree from Xi'an Jiaotong University, China in 1983 and Ph.D. degree from the University of Newcastle, Australia in 1990. From August 1989 to August 1991, he was a Postdoctoral Fellow at University of Adelaide. Since August 1991, he has been with School of EEE, Nanyang Technological University, where he is currently a Full Professor. His main research activities are in the areas of control systems and applications, intelligent power management system, intelligent mobile robots, cyber-

physical systems.

He is a Fellow of IEEE, a member of IEEE Fellow Committee from January 2011 to December 2013 and a Distinguished Lecturer of IEEE Control Systems Society from February 2010 to February 2013. He received the IES Prestigious Engineering Achievement Award 2005 from the Institution of Engineers, Singapore (IES) in 2005. He received the Best Paper Award of IEEE Transactions on Industrial Electronics in 2017.

He is an Associate Editor of a number of journals including *Automatica*, *IEEE Transactions on Industrial Electronics* and *IEEE Control Systems Magazine*. He is the Executive Editor-in-Chief, *Journal of Control and Decision*. He served the IEEE Transactions on Automatic Control as an Associate Editor from January 2000 to December 2002. He has been actively involved in organizing international conferences playing the roles of General Chair, General Co-Chair, Technical Program Committee Chair, Program Committee Member, General Advisor, Publicity Chair and so on.

Cheng-Leong Ng received the B.Eng and M.S. degree from National University of Singapore in 2006 and 2010 respectively. He also received the M.S. degree from Naval Postgraduate School, Monterey, California, U.S. He is currently working in ST Dynamics Pte Ltd as an Assistant Principal Engineer with a focus in multi robot systems.

Ying Zou received her B.Eng. degree in Measurement and Control Technology and Instrument from Central South University, Changsha, China in 2013 and M.Sc. degree in Instrument Science and Technology from Beihang University (BUAA), Beijing, China in 2015. Since then, she has been working toward the Ph.D. degree in School of Electrical and Electronic Engineering at Nanyang Technological University, Singapore.

Her research interests include adaptive control, nonlinear systems control, mobile robot control and multi-agent system control.